A Neural Associative Pattern Classifier

Francisco J. López Aligué[1], Isabel Acevedo Sotoca[1], Ignacio Álvarez Troncoso[1], Carlos García Orellana[1], and Horacio González Velasco[1]

[1] Universidad de Extremadura, Facultad de Ciencias. Departamento de Electrónica e Ingeniería Electromecánica.
Avda. de Elvas, s/n
06007 – BADAJOZ
Tfno.: 924 289 542
Fax: 924 289 543
E.mail: aligue@unex.es.

RESUMEN

En este trabajo, analizamos las características de la Memoria asociativa bidireccional (BAM) desde el punto de vista de los clasificadores de patrones. Para ello, estudiamos su comportamiento en función de la estructura neuronal que la soporta y de las posibles mejoras que admite. En concreto, los mejores resultados se han obtenido a través de la elección adecuada de los patrones de entrenamiento, de la adecuación de los umbrales y de la función de activación de las neuronas, lo cual se realiza de forma automática al principio de la programación de la red, quedando ya fijados para el resto del trabajo. Como prueba y comparativa, en una aplicación de clasificación de caracteres manuscritos, los resultados obtenidos indican claramente que el método propuesto conlleva una señalada mejora del rendimiento asumible en una BAM, llegándose a aciertos del 100% en el total de la base de datos NIST#19. También se ha probado esta red con otra base de datos, la bien conocida UCI, donde se ha Puesto de manifiesto la robustez del sistema y su resistencia frente a perturbaciones tales como ruidos aleatorios de hasta el 40%, lo que significaba la casi completa indistinguibilidad de los patrones de entrada.

PALABRAS CLAVE : Redes Neuronales, Memorias Asociativas, Reconocimiento de Patrones.

TÓPICOS : Redes Neuronales

SECCIÓN: Paper Track

OBSERVACIONES: Este trabajo NO ha sido enviado a ningún otro Congreso.

A Neural Associative Pattern Classifier

Francisco J. López Aligué[1], Isabel Acevedo Sotoca[1], Ignacio Álvarez Troncoso[1], Carlos García Orellana[1], and Horacio González Velasco[1]

[1] University of Extremadura, Dept. of Electronics, Faculty of Sciences. Av. Elvas s/n, 06071 Badajoz, Spain. aligue@unex.es.

Abstract. In this work, we study the behaviour of the Bidirectional Associative Memory (BAM) in terms of the supporting neural structure, with a view to its possible improvements as a useful Pattern Classifier by means of class associations from unknown inputs, once mentioned classes have been previously defined by one or even more prototypes. The best results have been obtained by suitably choosing the training pattern pairs, the thresholds, and the activation functions of the network's neurones, by means of certain proposed methods described in the paper. In order to put forward the advantages of these proposed methods, the classifier has been applied on an especially popular hand-written character set as the well-known NIST#19 character database, and with one of the UCI's data bases. In all cases, the method led to a marked improvement in the performance achievable by a BAM, with a 0% error rate.

1 Introduction

We present a design for a two-layer neural network, adapted from the conventional BAM structure, and specifically aimed at the classification of characters in the strong presence of noise or distortion. To achieve classification results that are completely error free, we defined a method of prototype selection in the training set, and a new formulation of the thresholds and the activity function of the neurons of the network. The classifier was tested on two widely accepted databases - NIST #19 and UCI. In all cases, even under unfavourable conditions, the success rate was 100%.

The application of associative memories to Pattern Recognition has been one of the most popular lines of research for some time now. Some significant theoretical studies in this field appeared in the sixties [1, 2, 3]. A network model was first established in work by Kosko [4] and showed a high degree of immunity to noise and distortions. After these first studies, there appeared operational neural models known as Willshaw models, which incorporated Hebbian learning in binary synapses and hetero-associative one-step retrieval [5]. There followed a number of developments aimed at improving the efficiency of these associative memories, with an analysis of all potential modifications of the elements of the process and of the retrieval methods [6].

Different training methods were described in the years following Kosko's original work to attempt to ensure the recognition of all the pairs presented to a BAM. Two encoding strategies were proposed in [7]: multiple training and dummy augmentation. Then the recovery of all the pairs was guaranteed using the multiple training method [8], and the same method was applied to the "back-propagation" algorithm for use in associative memories [9].

At around the same time, a method based on Housenholder transformations was proposed [10] to ensure that the pairs give rise to energy function minima. In doubling the number of connections, this method has the drawback that the evolution of the network may enter a cycle and not converge to a stable state. An improvement was later proposed in [11] which ensured convergence by avoiding such cycles. Two learning algorithms were proposed in [12] which improved Kosko and Wang's results. Amongst other later methods, we might cite that of Arthithan & Dasgupta [13] who approach the problem of the network's recognition of all the pairs and its behaviour with respect to noise. Other forms of altering the characteristics of a BAM may be found in the literature [14,15,16,17]. Recently, a three-layer feedforward implementation has been proposed [18] which guarantees secure recall for a determined number of associated pairs, and one finds in [19] one of the last attempts to increase a BAM's recall capacity. An extensive review of the state-of-the art of all the methods used up to now can be found in [20].

For the Associative Memories implemented by means of neural networks, there have been studies of methods of improving their capacity for storage and retrieval as a function of the neuron thresholds for the case of totally interconnected networks [21], for partially connected networks, which are closer to real biological structures [22], and even for such specific structures as cellular neural networks, [23, 24].

2 Architecture of the Classifier System

There are two clearly differentiated units in our system. On the one hand there is an optional geometrical-type preprocessor responsible for eliminating the topological components in cases in which the image aspect of the input character to be classified is important, as is the case with alphanumeric characters, certain kinds of images, etc. On the other hand, after this pre-processor, there is the neural network itself responsible for the definitive classification of the input.

Basically then, this is a two-layer, BAM-type associative memory, in which the inputs of each layer's neurons are totally connected to the outputs of the previous layer's neurons, but there exist no sideways connections within any given layer. The neurons of the first layer (the "input" layer in conventional BAM nomenclature) are defined by the expression

$$y_j = F(I_j) = F\left[\sum_{i=1}^{N} m_{ji} x_{ki} - \theta(\Theta_j)\right] \tag{1}$$

where "I_j" represents the excitation coming from the second layer's neurons. With respect to the latter, the main difference is that the threshold term "$\theta\,(\Theta_j)$" does not appear in their function. The significance of this will be seen below. Neither is the function F(.) the same. Whereas for the second layer, the usual sigmoid or the step functions are used, for the first we chose a multi-step function for the reasons that we will present below in discussing the "dead zone".

The network functions under the control of a supervisory unit which, when it detects that the input has been assigned to a class, halts the oscillating cycle that is characteristic of this type of memory.

This network is conceived for classification of its inputs, so that, unlike the typical BAM formulation, and even though it is a hetero-associative type of memory, the second character set is fixed beforehand and corresponds to what we shall denominate a set of "class" vectors $V_i = (v_{i1}, v_{i2}, ..., v_{i26})$, that belongs to a canonical structured set with equidistance 2 as in [25], so that one and only one of its components – that which indicates the numeral of the class – is "1" and the rest are "0":

$$V_1 = (1, 0, ..., 0); V_2 = (0, 1, 0, ..., 0); \; ... \; ; V_{26} = (0, 0, ..., 0, 1).$$

Thus, for example, in the case of handwritten character classification, we construct the pairs $\{A, V_1\}$, $\{B, V_2\}$, ... $\{Z, V_{26}\}$, then we obtain a new set of pairs $\{X_1, Y_1\}$, $\{X_2, Y_2\}$, ..., $\{X_{26}, Y_{26}\}$, where $\{X_i\}$ is the set of vectors obtained by bipolarization from the prototype set $\{A, B, C, ..., Z\}$ and $\{Y_i\}$ the bipolar vectors computed from the class vectors $\{V_1, V_2, V_3, ..., V_{26}\}$. We thus have the Relation Matrix constructed according to its original definition,

$$M = \sum_{i=1}^{N} X_i^T \bullet Y_i = \begin{pmatrix} m_{11} & \cdot\,\cdot & m_{1N} \\ \vdots & \ddots & \vdots \\ m_{P1} & \cdot\,\cdot & m_{PN} \end{pmatrix} = \left(m_1^T, .\,. m_N^T\right); \; m_j^T = \begin{pmatrix} m_{1j} \\ \cdot\,\cdot \\ m_{Pj} \end{pmatrix}. \tag{2}$$

due to the fact that one is dealing with class vectors, one can verify that the column vectors of the relation matrix have the following structure:

$$m_i^T = X_i^T + \sum_{j \neq i}^{N} X_j^{cT} \; . \tag{3}$$

where $\mathbf{X}_i^{cT}$ is the conjugate vector of $\mathbf{X}_i^T$.

At this point, let us suppose that a prototype $\boldsymbol{A_k}$, k = 1, ..., Q, is presented at the first layer, and A_k belongs to class "j". Since, as was stated above, the first layer has Q neurons with P inputs each, each neuron receives an excitation resulting from equation (4) which is:

$$I_j = \sum_{i=1}^{P} x_{ki} \cdot m_{ij} = \boldsymbol{x}_k \boldsymbol{m}_j = \boldsymbol{x}_k \Big(\boldsymbol{x}_j^T + \sum_{\substack{l=1 \\ l \neq j}}^{Q} \boldsymbol{x}_l^{cT} \Big). \qquad j = 1, \ldots Q, \tag{4}$$

This constitutes the complete expression that defines the excitation, which each neuron of a layer receives from the other layer. We shall see that this expression will be highly useful in the following section.

Figure 1 shows the block diagram for our classifier. The external input can be fed into the memory either directly or by way of the geometrical preprocessor.

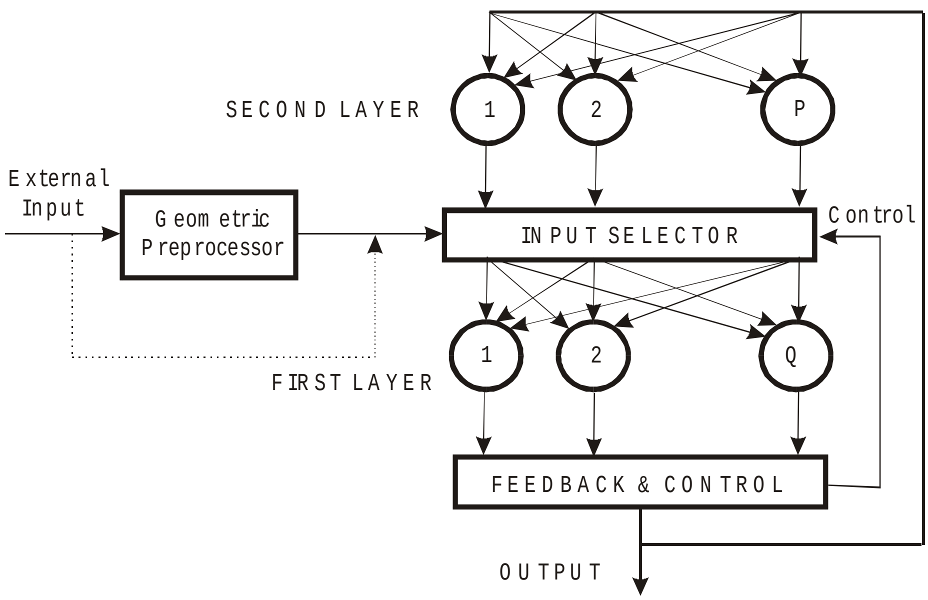

Fig. 1. Block diagram of the proposed pattern classifier.

For this purpose, we take advantage of the input selection unit which also serves to switch the signals that arrive at the first layer, selecting between the external input - the first activation of the BAM - and the outputs from the second layer - the normal running cycle.

In accordance with its definition, the first layer (so called the input layer) has N neurons, the same number as classes, and the second layer (the feedback layer) has P neurons, corresponding to the dimension of input characters to be classified. Each neuron of the first layer has P inputs, one for every output of the second layer, whereas the neurons of the second layer have N inputs, connected to the outputs of the N neurons of the first layer. With this arrangement, imposed by the classical definition of the BAM, the number of neurons and the inputs are perfectly known from the outset.

The remaining point is to decide the threshold θ (Θ_j) of the neural functions, as well as the particular form of the activity function $F(.)$. This will be the main objective of the following section.

3 The Adaptive Mechanism

The reason for this classifier's success resides in its adaptive process, which really begins with the selection of the training set. Indeed, the process of training the network is initiated with the selection of prototypes, with one prototype being chosen for each class. The choice is made by the method of greatest difference, seeking those characters, which provoke the greatest excitation I_j in the neuron of their class, and the least in the others. This process is performed under the control of a specific subprogram, and only takes place at the moment of choosing the classes that one wants to use. In this way, with all the prototypes not belonging to class "j", we have the maximum excitation at the j-neuron with $\boldsymbol{X}_k \in \boldsymbol{A}_k \neq$ Prototype $\boldsymbol{A}_j$,

$$Max\ I_k = Max\ \left(\sum_{i=1}^{P} w_{ij} x_{ki}\right). \tag{5}$$

which means that the maximum of the excitation coming from the prototypes of the class differs from the excitation for the prototypes of the remaining classes by at least a quantity that can be calculated as

$$\varepsilon_j = I_j - \text{Max}\ I_k . \tag{6}$$

We can now define the specific function of each of the first layer's neurons:

$$y_j = F(I_j) = F\left(\sum_{i=1}^{P} w_{ij} x_i - (\varepsilon_j - \chi)\right). \tag{7}$$

This replaces the initial definition (1). A "fudge" term χ has been introduced into this equation which avoids the possibility that there may at some moment appear an input of another class but of greater similarity to the prototype than those used in the initial calculation.

4 The Dead Zone

One of the other possible ways of increasing the networks' storage capacity is to alter the shape of the activity function. We shall now analyse the improvement in classical performance given by changing the activity function to a form similar to that of multi-step activation functions [26, 27] or even more similar to ternary neurons [28].

If the neurons are excessively sensitive, it may occur that any variation in the excitation provoked by a variation in the input alters sharply and unforeseeably the value of the output. We should therefore try to restrict, within certain limits for each class, the states of excitation to which the neurons should respond. It is a matter of endowing the neurons of the system with a variable degree of stability so that they are able to maintain the output constant against distorted inputs without having to completely reprogram the neural network, and so that the said neurons do not fire by mistake in response to inputs that are highly distorted or noisy.

In order to modulate the sensitivity of the neurons, we propose defining a safety margin for each class, to thereby ensure that the neurons responsible for the prototype classification do not generate outputs of value "+1" in response to the prototypes of the other classes. To this end, we choose the model of neuron function shown in figure 2:

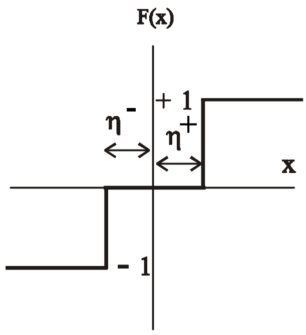

Fig. 2. The neuronal function with the dead zone term η.

With the introduction of the dead zone term into the neuron function, we get the neurons to fire within the limits learnt as the "allowed excitation limits of the class". In this way, the response to the presence of an unknown input to the network will be closer to the response that we have stored for the prototype of its class. With this premise, one can expect that the network will have a maximum degree of efficiency for pattern classification, since we have two ways to suppress the overlap noise and ensure convergence of the network – the threshold and the dead zone. The general output function for the first-layer neurons will now be put into the following form:

$$y_j = F\,(I_j,\,\eta_j) \;=\; \begin{cases} +1 & \text{if} \quad I_j > \eta_j^+ \\ 0 & \text{if} \quad \eta_j^- \le I_j \le \eta_j^+ \\ -1 & \text{if} \quad I_j < \eta_j^- \end{cases} \tag{8}$$

With max $\{I_i\} \le \eta_j^-$; min $\{I_j\} \ge \eta_j^+$; $i \neq j$ and where $\{I_i\}$ is the strongest excitation received from the prototypes of the other classes, as was defined in (5), while min $\{I_j\}$ is the minimum excitation provoked by some prototype of the class "j" in question. In this equation, F (.,.) is the neural function defined in equation (7) and η_j is each neuron's dead zone term.

5 Results

To evaluate our system's performance, we first chose one of the most widely used and universally accepted character sets, the NIST #19 (see [29]), handwritten character database, which contains binary images of 3699 forms filled in long-hand, and 814 255 alphanumeric characters taken from those forms. The isolated characters are 128x128 pixels in size, and are grouped into 62 classes which correspond to the upper and lower case letters of the English alphabet and the digits "0" to "9".

Table 1 lists the results obtained on the whole test set as the threshold is varied of the different neurons associated with each class. The table gives the correct classification results (%) in a rail-to-rail threshold scan on each group of characters of the NIST alphabet. The equation governing the threshold scan is $\theta = \theta_j + \Delta\theta$, where $\Delta\theta$ is expressed in hundreds. Thus the threshold of, for instance, the class "a" neuron varies between 17 384 (the value obtained from the character used as prototype) and 30 384 (= 17 384 + 13 000).

As a completely different case to study, which means that we do not handle inputs with a topologically characterized structure, we tested our improved BAM as Pattern Classifier on a database of the also well-known UCI [30]. As a simple test example, we chose the DISPLAY 1 database, consisting of a set of binary “characters” representing either the excitation state or extinction state of the lighting segments which form a common lighting diode-based display. The obtained results can be seen in Table 2.

Table 1. Classification success rate (in %) in a scan of the threshold value $\Theta_j = \theta_j + \Delta\theta$ (in hundreds) on characters of the NIST alphabet. Here θ_j is an a priori defined threshold value

Prot.	θ_j	$\Delta\theta$							
		0	20	40	80	90	100	115	130
a	17384	5	100	100	100	100	100	25	0
b	17820	0	10	100	100	100	100	45	0
c	17176	0	85	100	100	100	95	25	0
d	18056	0	90	100	100	100	100	50	0
e	18012	0	65	100	100	100	100	80	0
f	16096	0	35	100	100	100	100	100	15
g	16366	0	15	100	100	100	100	55	0
h	17548	0	0	100	100	100	100	90	0
i	16242	0	100	100	100	100	100	75	0
j	14370	0	95	100	100	100	100	0	0
k	18666	0	70	100	100	90	95	0	0
l	18646	20	100	100	100	100	100	65	0
m	18194	70	100	100	100	95	35	0	0
n	17298	20	100	100	100	100	60	0	0
o	16276	0	100	100	100	100	100	85	0
p	14992	0	65	100	100	100	100	100	0
q	16524	0	95	100	100	100	100	85	20
r	16406	0	90	100	100	100	100	100	90
s	16904	0	80	100	100	100	95	30	0
t	16134	0	0	100	100	100	100	95	0
u	15844	5	70	100	100	100	95	30	0
v	14244	0	75	100	100	100	100	70	0
w	15520	25	100	100	100	90	75	15	0
x	15976	0	45	100	100	95	50	0	0
y	16436	0	35	100	100	100	95	20	0
z	15636	25	100	100	100	95	90	55	0

Table 2. Success rates for the different noise inputs for the DISPLAY 1 database . Note that a noise around 40% means a variation of the statement of the led set, over a half of all them, that leads to a misclassification due to the fact that coincidences between inputs become unsolvable.

Stochastic Noise	Classification Success
0	100
10	100
30	85
40	78

For further comparation purposes, in literature [see 31, 32] can be found the obtained results for this database using several techniques of Pattern

Classification:

1.- With 200 training and 5000 test instances:

- Optimal Bayes classification rate:	74%
- CART decision tree algorithm: (resubstitution estimate)	71%
- Nearest Neighbour Algorithm:	71%

2.- With 2000 training and 500 test instances:

- C4 decision tree algorithm: (using pessimistic pruning)	72.6%

3.- With 400 training and 500 test cases

- IWN system: (using the And-OR classification algorithm)	73.3%

The results on those databases were not the only ones, however. We also used the BAM with the threshold and dead zone adaptations described with another type of characters (radiographic images) with the same success rate.

Acknowledgements

The authors wish to express their gratitude to the Spanish C.I.C.Y.T. and to the Junta de Extremadura for their support of this work through grants TIC2001-0881 and 2PR01A007, respectively.

References

1. Steinbuch, K.: Die lernmatrix, *Kybernetik,* **1** (1961) 36 – 45
2. Anderson, F. A.: A simple neural network generating an inter-active memory, *Mathematical Biosciences.***14** (1972) 197-220
3. Kohonen, T.:. *Associative memory: A system-theoretical approach,* Springer Verlag, New York (1977)
4. Kosko, B.: Bidirectional associative memories, *IEEE Trans. on Systems, Man and Cybernetics,* **18**(1) (1988) 49-60
5. Willshaw, D. J., Buneman, O. P., Longuet H. C.: Nonholographic associative memory, *Nature,* **222** (1969) 960-962. Reprinted in Anderson & Rosenfeld (eds.), *Neurocomputing: Foundation of research,* MIT Press, Cambridge, (1988)
6. Sommer, F. T., Palm, G.: Improved bidirectional retrieval of sparse patterns stored by Hebbian learning, *Neural Networks,* **12** (2) (1999) 281-297
7. Wang, Y. F., Cruz, J. B., Mulligan, J. H.: Two coding strategies for bidirectional associative memory, *IEEE Trans. on Neural Networks*, **1**(1) (1990) 81-82
8. Wang, Y. F., Cruz, J.B., Mulligan, J. H.: Guaranteed recall of all training pairs for bidirectional associative memory, *IEEE Trans. on Neural Networks*, **2**(6) (1991) 559-567
9. Wang, Y. F., Cruz, J.B., Mulligan, J. H.: Multiple training concept for back-propagation neural-networks for use in associative memories, *Neural Networks*, **6**, (1993) 1169-1175
10. Leung, C. S., Cheung, K. F.: Householder encoding for discrete bidirectional associative memory, *Proc. of the IEEE Intl. Joint Conf. on Neural Networks; IJCNN'91*, **1**, (1991) 124-137
11. Leung, C. S.: Encoding method for bidirectional associative memory using

projection on convex sets, *IEEE Trans. on Neural Networks*, **4**(5) (1993) 879-881
12. Zhuang, X. Huang, Y., Chen, S-S.: Better learning for bidirectional associative memory, *Neural Networks*, **6** (1993) 1131-1146
13. Arthithan, G., Dasgupta, C.: On problem of spurious patterns in neural associative memory models, *IEEE Trans. on Neural Networks* **8**(6) (1997) 1483-1491
14. Jeng, Y. J., Teh, C. C.: Modified intraconnected bi-directional associative memory, *IEE Electronics Letters*, **27**(20) (1991) 1818-1819
15. Lee, D. L.: Generalized intraconnected bidirectional associative memories, *IEE Electronics Letters*, **34**(8) (1998) 736-738
16. Heekuck, O. H., Kothari, S. C.: Adaptation of the relaxation method for learning in bidirectional associative memory, *IEEE Trans. on Neural Networks*, **5**(4) (1994) 576-583
17. Simpson, P. K.: Higher-ordered and intraconnected bidirectional associative memories, *IEEE Trans. on System Man and Cybernetics 2*, **20**(3) (1990) 637-653
18. Park, J, Kim, H.-Y., Park, Y., Lee, S.-W.: A synthesis procedure for associative memories based on space-varying cellular neural networks, *Neural Networks,* **14** (1) (2001) 107-113
19. Müller, P., Rios, D.: Issues in Bayesian analysis of neural network models. *Neural Computation,* **10** (1995) 571-592
20. Arica, N., Yarman-Vural, T.: An overview of character recognition focused on off-line handwriting, *IEEE Trans. on Systems, Man and Cybernetics — Part C,* **31**(2) (2001) 216-233
21. Wu, Y., Pados, D. A.: A feedforward bidirectional associative memory, *IEEE Trans. on Neural Networks,* **11**(4) (2000) 859-866
22. Nadal, J. P., Toulouse, G.: Information storage in sparsely coded memory nets, *Network.* **1** (1990) 61–74
23. Buckingham, J., Willshaw, D.: On setting unit thresholds in an incompletely connected associative net, *Networks,* **4** (1993) 441-459
24. Chua, L. O., Yang, L.: Cellular neural networks: Theory and applications, *IEEE Trans. on Circuits and Systems,* **35** (1988) 1257-1290
25. Kumar, S.: Memory annihilation of structured maps in bi-directional associative memories, *IEEE Trans. on Neural Networks*, **11**(4) (2000) 1023-1038
26. Crespi, B.: Storage capacity of non-monotonic neurons, *Neural Networks,* **12** (10) (1999) 1376-1389
27. Alippi, C., Piuri, V., Sami, M.: Sensitivity to errors in artificial neural networks: A behavioral approach. *Proc. of the IEEE International Symposium on Circuits and Systems*, (1994) 459-462
28. Bollé, D., Domínguez, D. R. C., Amari, S. I.: Mutual information of sparsely coded associative memory with self-control and ternary neurons, *Neural Networks,* **13** (4-5) (2000) 455-462
29. NIST Special Data Base #19: Image Processing Group. Advanced Systems Division. National Institute of Standards and Technology (1995)
30. Merz, C.J., and Murphy, P. M.:UCI repository of machine learning databases. (1998)
31. Quinlan, J.R. : Simplifying Decision Trees. *In International Journal of Man-Machine Studies,* (1987)
32. Tan, M., Eshelman, L.: Using Weighted Networks to Represent Classification Knowledge in Noisy Domains, in Ann Arbor, Michigan: Morgan Kaufmann (eds.), *Proc. of the 5th Intl. Conf. on Machine Learning* (1988) 121-134